数理解析研究所講究録
第 833 巻 1993 年 98-109

グラフの多重目標点分離問題に対する近似アルゴリズムについて

京都大学数理工学科
前田 尚久 (Naohisa Maeda)
永持 仁 (Hitosi Nagamochi)
茨木 俊秀 (Tosihide Ibaragi)

1 序論

ネットワーク (G, w) を考え，このネットワークを指定された要求を満たすいくつかの成分に分ける重み最小の分割を見つける問題について考える．

このような問題の中で，ネットワーク $(G=(V,E),w)$ と分割数 $k \geq 2$ が与えられた時，G を k個の成分に分割する辺の部分集合を考える．このような辺の部分集合のうちで，その辺の重みの和が最小であるものを求める問題は多重分離問題 (Multiway Split Problem) と呼ばれる．これに対し，G が目標点と呼ばれる k個の点を持ち，G をこれらの点を一つずつ含む k個の成分に分割する辺の部分集合を考える．そのような辺集合のうち重みの和が最小のものを求める問題は多重カット問題 (Multiway Cuts Problem) と呼ばれる．

このうち多重分離問題は kを問題のサイズの一つと考えると，NP-困難になることが知られている [3] が，最適解を得るための$O(n^{k^2/2-k+11/2})$時間（これは，kを定数とみなせば多項式時間である）のアルゴリズムが得られている [3]．また，$k=3$ かつグラフが平面のときはさらに高速なアルゴリズムが設計されている [4]．H. Saran と V. V. Vazirani は最適解の $2(1-1/k)$ 倍以内の重みを持つ分割を $O(n \cdot M(n,m))$ 時間で構成するアルゴリズムが提案している ([7])．ただし，$M(n,m)$ は n 節点，m 辺のネットワーク上で最大流問題を 1 つ解くための時間である．多重カット問題は，一般のグラフにおいて $k=3$ とした場合，あるいは平面グラフに限っても kをその問題のサイズの一つとした場合には NP-困難であることが示されている [1]．多重カット問題に対しても最適解の $2(1-1/k)$ 倍以内の重みを持つ分割を $O(k \cdot M(n,m))$ 時間で構成するアルゴリズムが提案されている [1].

本研究では，ネットワーク $(G=(V,E),w)$，目標点 $T \subseteq V$，分割数 $k \leq |T|$ が与えられた時，ネットワークを少なくとも 1 個の目標点を持つ k個の成分に分ける分割を求める多重目標分離問題を新たに導入する．この問題は $T=V$であれば多重カット問題，$k=|T|$ であれば多重分離問題と一致する．そして多重目標点分離問題に対して，最

適解の $2(1-1/k)$ 倍以内の重みをもつ分割を求めるアルゴリズムを提案する.

2 定義

$G=(V,E)$ を節点集合 V および辺集合 E を持つ単純無向グラフ (非連結グラフであってもよい) とする. 節点 x と y を端点とする辺を (x,y) と書く.

(G,w) をグラフ G の各辺 e に正の実数値の重み $w(e)$ を持つネットワークとする.

(1) **分割**; V の分割 $\{V_1,V_2,\cdots,V_k\}$ は同値関係 R を用いて表す. 即ち,

$$V/R=\{V_1,\ldots,V_k\}$$

となる同値関係 R を考える. ただし, V/R は V の R による商集合である. 本研究を通じて, R を分割, $W\subseteq V$ に対して W/R を W の R による分割, $V_1,\ldots,V_k$ を V の分割 R による成分と呼ぶ. また,

$|V/R|=2$ である V 上の分割 R をとくに V のカットと呼ぶ.

$|V/R|=k$ なる V 上のすべての分割 R の集合を $\Omega(k,V)$ で表す.

(2)**2 節点の分離**; 2 節点 x,y がそれぞれ異なる分割の成分に含まれるとき, つまり,

$$x\in V_i,\ y\in V_j\ (i\neq j)$$

であるとき, x と y とは分割 R によって分離されるという.

(3) **分割の重み**; V の分割 R による成分の集合 $V/R=\{V_1,\ldots,V_k\}$ に対し $E(R)$ をその両端点が異なる成分に含まれる辺の集合, $w(R)$ を分割 R の重みとする. 即ち,

$$E(R)\equiv\{(u,v)\in E\mid u\in V_i,\ v\in V_j, i\neq j$$

$$w(R)\equiv\sum_{e\in E(A)}w(e)$$

と定義する.

(4) **分割の和**; V 上 の い く つ か の 分 割 $R_1,R_2,\ldots,R_k$ に対して

$$xRy\leftrightarrow(xR_1y)\wedge(xR_2y)\wedge\cdots\wedge(xR_ky)$$

により定義される V 上の分割を簡単に

$$R_1\wedge R_2\wedge\cdots\wedge R_k\quad ,\ \text{あるいは}\ \bigwedge_{h=1}^{k}R_h$$

と記す.

(5) **分割の強弱**; R_1 と R_2 を V 上の 2 つの分割とする. このとき, 任意の $x,y\in V$ に対して,

$$xR_1y\quad \text{ならば}\quad xR_2y$$

であるとき, R_1 は R_2 より強い (R_2 は R_1 より弱い) と言う.

3 問題

最小多重目標点分離問題とはネットワーク $(G=(V,E),w)$ において, 目標点の集合 $T\subseteq V$ と $2\leq k\leq|T|$ をみたす分割数と呼ばれる整数 k が与えられたとき

$$|T/R|\geq k$$

を満たす V 上の分割 R で重み $w(R)$ を最小にするものを求める問題である．

この問題は，特に，$T = V$ の場合は多重分離問題と，$k = |T|$ の場合には多重カット問題と一致する．

本研究では TARGET, SPLIT という二つの近似アルゴリズムを考え，これらのアルゴリズムによって得られる分割の重みが，最適値の $2(1-1/k)$ 倍以内になることを示す．

4 アルゴリズム

4.1 アルゴリズム TARGET

ここでは多重分離問題に対する H. Saran と V. V. Varizani の近似アルゴリズム [7] を拡張した多項式アルゴリズムを示し，それが最小多重目標点分離問題に対する近似解を求めることを示す．

アルゴリズム　TARGET

1. $$L : A_1, A_2, \ldots \qquad (1)$$
   を V のカット（$|V/A_i| = 2$ なる分割）の集合 $\Omega(2,V)$ の重みの非減少順の列挙とする．

2. 便宜上 A_0 を V 自身を 1 つの分割の成分とする（すなわち，$|V/A_0| = 1$ なる）分割とする．
   $|T/\bigwedge_{h=0}^{i} A_h| > |T/\bigwedge_{h=0}^{i-1} A_h|$ なる添え字 i の集合を $I(L)$ とする．

3. ステップ 2 で得られた $I(L)$ を $\{i_1, i_2, \ldots, i_p\}$ $(i_1 < i_2 < \cdots < i_p)$ として，
   分割 $A_L = \bigwedge_{h=i_1}^{i_p} A_h$ を出力し終了． □

4.2 TARGET の性能

TARGET により得られた $I(L)$ は次を満たす．

任意の $i_j \leq h < i_{j+1}$ $(i_j, i_{j+1} \in I(L))$ なる k に対して，$I(L)$ の作り方より明らかに，

$$T/\bigwedge_{h=1}^{h} A_h = T/\bigwedge_{h=i_1}^{i_j} A_h. \qquad (2)$$

ここで，

$$n(1) \equiv 1,$$

$$n(j) \equiv |T/\bigwedge_{h=i_1}^{i_{j=1}} A_h| \quad (j \geq 2)$$

と定義する．

定理 4.1 $n(l-1) < k \geq n(l)$ なる l に対し，分割 $\bigwedge_{h=i_1}^{i_l} A_h$ は最適解の $2(1-\frac{1}{k})$ 倍以下の重みを持つ．□

(証明) 最小目標点分離問題に対する最小重みの分割 B^* を選び，
$V/B^* = \{V_1, V_2, \ldots, V_k\}$
とおく．ここで，各 V_j に対し，$V/B_j^* = \{V_j, V - V_j\}$ なるカット $B_j^* \in \Omega(2,V)$ を考える．(1) における $\Omega(2,V)$ の重みの非減少順の列挙 $L : A_1, A_2, \ldots$ に対し，

$$V/A_{i_j^*} = V/B_j^* = \{V_j, V - V_j\}$$

を満たす添え字 i_j^* の集合を $I^*(L) = \{i_j^* \mid j = 1, 2, \ldots, k\}$ とする．

$|T \cap V_j| = 1$ $(1 \geq j \geq k)$ より明らかに

$$|T/\bigwedge_{h=i_1^*}^{i_j^*} A_h| = j + 1 \quad (1 \leq j \leq k-1)$$

さて，$I(L), I^*(L)$ に対し，上で定めた $\{n(j) \mid j = 1, \ldots, l\}$ をもちいて

$$|T/\bigwedge_{h=1}^{j})A_{i_h}| \geq n(j)+1,$$

$$|T/\bigwedge_{h=1}^{n(j)} A_{i_h^*}| = n(j)+1 \tag{3}$$

が成り立つことに注意する．ここで，明らかに，

$$|T/\bigwedge_{h=1}^{n(j)} A_{i_h^*}| \leq |T/\bigwedge_{h=1}^{i^*_{n(j)}}|$$

である．ここで，$i_j > i^*_{n(j)}$と仮定すると (2) より

$$\begin{aligned} |T/\bigwedge_{h=1}^{n(j)} A_{i_h^*}| &\leq |T/\bigwedge_{h=1}^{i^*_{n(j)}} A_h| \\ &\leq |T/\bigwedge_{h=1}^{j-1} A_{i_h}| \\ &= n(j) \end{aligned}$$

を得るが，これは (3) に矛盾する．従って次の補題を得る．

補題 4.1

$i_j \leq i^*_{n(j)}$ $(1 \leq j \leq l)$

□

つまり，L 上で $I(L) = \{i_1,\ldots,i_l\}$ の分布と $I^*(L) = \{i_1^*,\ldots,i_{k-1}^*\}$ の分布を考えたとき，A_{i_j} が対応する $A_{i^*n(j)}$よりも列挙 L において早く現れる．

補題より，各 $j=1,2,\ldots,l$に対し

$$w(A_{i_j}) \leq w(A_{i^*_{n(j)}})$$

を得る．従って

$$\begin{aligned} & w(A_{i_1}) + w(A_{i_2}) + \cdots + w(A_{i_l}) \\ \leq\ & w(A_{i^*_{n(1)}}) + \cdots + w(A_{i^*_{n(k-1)}}) \\ \leq\ & w(A_{i_1^*}) + \cdots + w(A_{i^*_{k-1}}) \end{aligned} \tag{4}$$

が成り立つ．これにより，TARGET によって得られる分割 $A_L = A_{i_1} \wedge \cdots \wedge A_{i_l}$ は最適解 B^*の $2(1-\frac{1}{k})$ 倍以下の重みを持つことを証明する．

$$\begin{aligned} w(A_L) &\leq \sum_{j=1}^{l} w(A_{i_j}) \\ &\leq \sum_{j=1}^{k-1} w(A_{i_j^*}) \quad \text{((4) より)} \\ &= 2w(B^*) - w(A_{i_k^*}) \\ &\leq 2(1-\frac{1}{k})w(B^*). \end{aligned}$$

以上より定理 1.1 の証明が得られた．

4.3 TARGET の実現

TARGET のステップ 1 において実際に全てのカットを列挙すれば $O(2^n)$ の手間がかかってしまう．以下では，TARGET に本質的に必要なカットだけをあらかじめ計算しておけば，ステップ 1 で全てのカットを列挙しなくても TARGET が望みの出力が多項式時間で得られることを示す．

$\Omega_{x,y}$を $(G=(V,E),w)$ において x と yとを分離するカット Rで重みが最小のカットの集合とする．各 x,yに対し，任意のカット $C_{x,y} \in \Omega_{x,y}$を一つ選んだ集合 $\{C_{x,y} \mid x,y \in T\}$ を考える．

補題 4.2 $L_T : C_1, C_2, \ldots$ を $\{C_{x,y} \mid x,y \in T\}$ の任意の重み非減少順の列挙とする．このとき，ステップ 1 でこの L_Tを用いて得られる出力と同じ出力を与える全てのカット$\Omega(2,V)$ の列挙 L が存在する．

(証明) 補題を示すためには，どんな$\Omega(2,v)$ の列挙 L に対しても TARGET によって得られるカット A_i $(i \in I(L))$ は適当な二点 $x,y \in T$に対し $A_i \in \Omega_{x,y}$を満たすことを示せばよい．ある

$\Omega(2,v)$ の列挙に対しアルゴリズムによって得らるあるカット $A_i(i \in I(L))$ を考える．このとき，$I(L)$ の作り方より，分割 $R_{i-1} : A_1 \wedge A_2 \wedge \cdots \wedge A_{i-1}$ の同じ成分に含まれ，A_iによって分離される二点 u,vが存在するはずである．$\Omega_{u,v}$の任意のカット $C_{u,v}$ も

$$|T/(R_{i-1} \wedge C_{u,v})| > |T/R_{i-1}|$$

を満たすので，L が重みの非減少順に並んでいることから，

$$w(A_i) \le w(C_{u,v})$$

である．即ち，A_iも$\Omega_{u,v}$に属する． □

ネットワーク $(G=(V,E),w)$ に対する $W \in V$ を張る Gomory-Hu 木とは次の二つの性質を満たす重み付きの木 $((W,F),c)$ である．

(1) 各節点対 $u,v \in W$間の (W,F) 上の唯一の道に含まれる枝の重みの最小値が u,vを分離する最小カットの大きさを表す．

(2) さらに，(W,F) からその最小値を与える枝を取り除くことによってできる 2 つの連結成分が，u,vを分離する最小カットの成分を表す．

補題 4.3 $W \in V$を張る Gomory-Hu 木は常に存在し，$|W|-1$ 回最大流問題を解くことにより求められる． □

Gomory-Hu 木 $((W,F),c)$ から 1 本の枝 e を取り除けば木は 2 個の連結成分に分かれ，これらを成分に持つカット R_eが定義できる．R_eの重みを $c(e)$ とする．このとき $\{R_e | e \in F\}$ を用いて，アルゴリズム TARGET を次のように書き直すことができる．

アルゴリズム TARGET'

1. 目標点Tを張る Gomory-Hu 木 $((T,F),c)$ を構成する．$\{R_e | e \in F\}$ の重み $(c(e))$ の非減少順の列挙を $L_{GH} : R_1, R_2, \ldots, R_{|T|-1}$ とする．

2. 便宜上 A_0を V自身を 1 つの分割の成分とする（すなわち，$|V/A_0| = 1$ なる）分割とする．

$$|T/\bigwedge_{h=0}^{i} A_h| > |T/\bigwedge_{h=0}^{i-1} A_h|$$

なる添え字 i の集合を $I(L)$ とする．

3. ステップ 2 で得られた $I(L)$ を $\{i_1, i_2, \ldots, i_p\}$ $(i_1 < i_2 < \cdots < i_p)$ として，
分割 $A_L = A_{i_1} \wedge A_{i_2} \wedge \cdots \wedge A_{i_p}$を出力し終了．
□

Gomory-Hu 木の性質より，各 $x,y \in T$を分離するカットで重み最小のものの 1 つはカット R_e で表されるので．列挙 L_{GH}は補題 4.2 で考えた列挙 L_Tの中の 1 種である．よって補題 4.2 より，TARGET' で得られる出力は TARGET でも得られる．よって，TARGET' により得られる分割は最適解の $2-1/k$倍以下の重みを持つ．

Gomory-Hu 木を利用することにより，TARGET' は $O(|T|M(n,m))$ 時間時間で実行することができる．($M(n,m)$ は n 節点，m 辺からなるネットワーク上で最大流問題を 1 つ解くための時間で，現在 $O(mn\log(n^2/m))$ 時間のものが知られている [2]）．

4.4 アルゴリズム SPLIT

文献 [7] にはもう 1 つの近似アルゴリズムが提案されている．これも以下のように多重目標点問題に拡張することができる．

アルゴリズム SPLIT

1. Vを1つの分割の成分とする分割をD_0とする（すなわち，$|V/D_0|=1$）；$(G_1,w_1):=(G,w)$.

2. $i=1,2,\ldots,k-1$ に対して以下を繰り返す.

   1. $T/\bigwedge_{h=0}^{i-1}D_h\{T_1,\ldots,T_i\}$ において，ある $1\le h\le i$ に対し

   $$|T_h/D|=2$$

   $$|T_j/D|=1 \quad (j\neq h)$$

   即ち,

   $$|T/(\bigwedge_{h=0}^{i-1}D_h)\wedge D| = |T/\bigwedge_{h=0}^{i-1}D_h|+1$$

   が成立するカット Dの中で (G_i,w_i) における重み最小のものを D'_hとする. $D'_h(1\le h\le i)$ の中でさらに重みの最小のものを D_iととする.

   2. (G_i,w_i) から $E(D_i)$ の辺を取り除いたネットワークを (G_{i+1},w_{i+1}) とする. □

4.5 SPLIT の性能

定理 4.2 *SPLIT*によって得られる分割 $D^*=D_1\wedge\cdots\wedge D_{k-1}$の重みは最適な分割 B^*の重みの $2(1-\frac{1}{k})$ 倍以下である.

（証明）$V/B^*=\{V_1,V_2,\ldots,V_k\}$ としたとき，カット B_j^*を $V/B_j^*=\{V_j,V-V_j\}$ であるカット，また集合Ω_{B^*}を$\Omega_{B^*}=\{B_1^*,B_2^*,\ldots,B_k^*\}$ であるとする. また，$w_i(R)$ を (G_i,w_i) におけるカット Rの重みとする. このとき,

$$w(D^*)=w_1(D_1)+\cdots+w_{k-1}(D_{k-1})$$

である. また，明かに $w_i(B_j)\le w(B_j)$ である.

以下のようにして，$j=k,k-1,k-2,\ldots,1$ の順に添字 $j(1\le j\le k)$ を決定して，集合Ω_{B^*}の列挙 L_{B^*} : $B_1,B_2,\ldots,B_k$を作れば，すべての $j(1\le j\le k-1)$ に対して $w_j(D_j)\le w_j(B_j)$ が成立する.

1. $j=k$について，Ω_{B^*}に含まれるカットのうち，重み最大のものを B_kとする.

2. $k-1\ge j\ge 1$ について,

   (G_j,w_j) において，D_jを決定するときのことを考える. Ω_{B^*}より $B_k,B_{k-1},\ldots,B_{j+1}$を取り除いた集合を$\Omega_{B_j^*}$とする.

   $$|T/\bigwedge_{B_j^*\in\Omega_{B_j^*}}B_j^*|=j+1$$

   $$|T/(D_1\wedge\cdots\wedge D_{j-1})|=j$$

   より，$\Omega_{B_j^*}$の中には $D_1\wedge\cdots\wedge D_{j-1}$よりも弱くないカットが存在する. ここで，これを B_jとすると

   $$w_j(B_j)\ge w_j(D_j) \quad となる.$$

□

以上のことより

$$\begin{aligned} & w(B_1)+\cdots+w(B_{k-1}) \\ \ge\ & w_1(B_{j-1})+\cdots+w_{k-1}(B_{j_{k-1}}) \\ \ge\ & w_1(D_1)+\cdots+w_{k-1}(D_{k-1}) \\ =\ & w(D^*) \end{aligned}$$

すなわち，定理を得る. □

4.6 SPLIT の実現

SPLIT のステップ 2 は，T に対する (G_i, w_i) 上の Gomory-Hu 木 [5] を構成すれば $O(|T|)$ 回の最大流問題を解く時間で実行することができる．全体として SPLIT は $O(k|T|M(n,m))$ 時間で実行することができる．ただし，$T = V$ の場合には SPLIT の計算において (G', w') の各成分の最小カットは最小カットアルゴリズムを使って求められるので，[7] の $O(mn + n^2 \log n)$ 時間の最小カットアルゴリズムを使えば $(k(mn + n^2 \log n))$ 時間で SPLIT を実行することができる．

5 数値実験

5.1 試験問題の生成

本実験では，ネットワークの節点数，目標点数や分割数などの異なる試験問題に対してアルゴリズム TARGET, SPLIT を適用して得られる分割の大きさや，実行時間の違いを調べてみた．実験では節点数と密度 (完全グラフに対する辺数の比) を与えて試験問題をランダムに発生させ，各々に対してアルゴリズムを適用して比較を行った．試験問題は以下のようにして発生させた．

1. 節点数 n, 密度 d, 目標点数 t，分割数 kを与える．
2. n 個の節点 $v_1, v_2, \ldots, v_n$を作る．
3. 全ての節点間に確率 $d\%$で重さ wの枝 (枝の重さ wは 1 から 100 までの整数とする) を張り，ネットワーク (G, w) を作る．
4. 節点 $v_1, v_2, \ldots, v_t$を目標点とする．

5.2 アルゴリズム OPT

アルゴリズム TARGET および SPLIT は共に最適な分割の 2 倍以下の分割を求めることは示したが，実際にどの程度最適な分割との差があるのかを調べるためには最適な分割を求めそれとの比較が必要である．ここでは最適な値を求めるアルゴリズムを示す．

アルゴリズム OPT

1. 各節点を分割数と同じ数の成分に分けるすべての組み合わせを求める．
2. 目標点の入っていない成分がある組み合わせを除き，その端点が異なる成分に含まれるような枝の重みを合計しその組み合わせによる分割の重みを求める．
3. ステップ 2 で求めた分割の重みの中で最小のものを最適な分割の重みとして出力する．

5.3 OPT と TARGET および SPLIT との比較

アルゴリズム TARGET 及び SPLIT によって得られる分割の重さは最適なものの 2 倍以内であることは証明したがそれらの値が実際にどの程度正しいかを調べてみた (表 1, 図 1)．節点数が 14 以上の場合についてはアルゴリズム OPT の計算時間が大きすぎて比較を行えなかった．

表の数字は各節点数ごとに 5 種類の異なるネットワークを作りアルゴリズムによって得られた分割の重みをアルゴリズム OPT によって得られた最適値との誤差を百分率で表した．0 は全く同じ

値を，0.0 は異なる値を出したものの誤差が 1%以下であったことを示す．

5.4 TARGET と SPLIT との比較

どちらのアルゴリズムを用いても枝が密なグラフに対しては全く同じ分割が得られることが多い．これは密なグラフの最小の二分割は一点とその他の点とを分ける分割であることが多いからだと思われる．実際にアルゴリズムを適用した後の分割を見てみると多くの（ほとんどの場合はただ一つの成分を除いて全ての）成分は節点を一つしか含まないことが多い．

そこでここでは密度の低いグラフについての比較を重点的に行った．実験は節点数，密度，目標点数，分割数の同じ問題を 5 問作りそれぞれに TARGET と SPLIT を適用して得られた分割の大きさと実行時間についての比較を行った．

(1) 目標点数，分割数を固定し節点数を変化させた場合を表 2, 図 2 に示す．

以下で TARGET および SPLIT はそれぞれのアルゴリズムによる値を示す．

TARGET および SPLIT ともに実行時間は節点数の増加にともない大きくなっているがどちらの実行時間も大差なくほぼ同程度となっているこれは分割数が小さいためである．

(2) 節点数，分割数を固定し目標点数を変化させた場合を表 3, 図 4 に示す．

TARGET および SPLIT 共に実行時間は目標点数によって大きく変化するが，どちらの実行時間も大差なくほぼ同程度となっているこれは分割数が小さいためである．

(3) 節点数，目標点数を固定し分割数を変化させた場合を表 4, 図 4 に示す．

分割数によって TARGET の実行時間は余り変化しないが SPLIT は分割数によって実行時間が大きく変わる．これは実行時間が TARGET は $O(|T|M(n,m))$ 時間 SPLIT は $O(k|T|M(n,m))$ 時間であるためである．

(4) 結論

以上の結果より

分割の大きさについては SPLIT のほうが良い値であることが多いものの，どちらのアルゴリズムを用いても殆ど違いはない．

実行時間については分割数が小さいと SPLIT の方が多少速いものの，大きな分割数に対しては TARGET の方がかなり速くなる．これは TARGET ではただ 1 回しか Gomory-Hu 木を張らないのに対して SPLIT では $k-1$ 回 Gomory-Hu 木を張るからである．

以上のことから，TARGET の方が SPLIT よりも実用に適していると思われる．しかし，この実験においては SPLIT でも最小の 2 分割を求めるのに Gomory-Hu 木を用いたが，その代わりに [6] のアルゴリズムを用いれば SPLIT の実行時間についてより良い結果が得られていたと思われる．

6 まとめ

本研究では，ネットワークの分割問題において，これまで別個に扱われていた多重分離問題と多重

カット問題とを，多重目標点分離問題を考えて，同時に扱えるようにした．そして，多重分離問題で提案されていた最適解の２倍以下の重みを持つ分割を求める近似アルゴリズム拡張して，多重目標点分離問題で最適解の２倍以下の重みを持つ分割を求める近似アルゴリズムを提案した．そして，他の分割問題に対しても，同様のアプローチで最適解の２倍以下の近似アルゴリズムを作るのは容易であると思われる．

今後の課題としては，分割問題に対して別のアプローチを考え，２倍よりも良い近似解を求める近似アルゴリズムを作ることを考えている．

参考文献

[1] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour and M. Yannakakis; "The complexity of the multiway cuts," *Proceedings of the 24th ACM Symposium on Theory of Computing*, Victoria, Canada, 1992, pp.241-251.

[2] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum flow problem," *Journal of the ACM*, 35, 1988, pp.921-940.

[3] O. Goldschmidt and D. S. Hochbaum, "Polynomial algorithm for the$k$-cut problem," *Proceedings of 29th Symposium on Foundations of Computer Sicence*, Los Angels, Calif., 1988, pp.444-451.

[4] D. S. Hochbaum and D. B. Shimoys, "An $O(|V|^2)$ algorithm for the planar 3-cut problem," *SIAM J. Algebraic and Discrete Methods*, 6, 1985, pp.707-712.

[5] T. C. Hu; *Integer programming and Network Flows*, Addison-Wesley Publishing Company, 1969

[6] H. Nagamochi and T. Ibaraki, "Computing edge-connectivity in multigraphs and capacitated graphs," *SIAM J. Disc. Math.*, 5, 1992, pp.54-66.

[7] H. Saran and V. V. Vazirani; "Finding k-cuts within twice the optimal, " *Proceedings of 32nd Symposium on Foundations of Computer Sicence*, San Juan, Puerto Rico, 1991, pp.743-751.

表 1: 目標点数= 節点数，分割数= 3，密度= 10%

節点数		5	6	7	8	9	10	11	12	13
分割の重み	TARGET	2	0.0	1	0.0	1	1	0	1	1
の誤差 (%)	SPLIT	1	0	0	1	0	1	0	0	0
実行時間	OPT	0.02	0.08	0.33	1.24	4.56	16.4	66.3	210.3	738.6

表 2: 目標点数= 50，分割数= 3，密度= 5%

節点数		50	100	150	200
分割の重みの比	SPL/TRG	0.999	0.999	1	1
実行時間	TARGET	7.540	31.098	71.688	126.09
	SPLIT	7.154	29.958	72.326	117.11

表 3: 節点数= 50，分割数= 3，密度= 5%

目標点数		3	6	9	10	20	30	40	50
分割の重みの比	SPL/TRG	1	0.999	1	1	1	1	1	0.999
実行時間	TARGET	0.280	0.704	1.182	1.302	2.870	4.414	6.054	7.560
	SPLIT	0.142	0.542	1.078	1.208	2.486	4.486	6.266	7.780

表 4: 節点数= 50，目標点数= 3，密度= 5%

分割数		3	4	5	6	7	8	9
分割の重みの比	SPL/TRG	0.999	0.999	0.998	0.995	0.995	0.997	0.994
実行時間	TARGET	7.562	7.600	7.522	7.518	7.530	7.520	7.514
	SPLIT	7.768	14.984	21.572	27.842	33.714	39.216	44.228

分割数		10	15	20	25	30
分割の重みの比	SPL/TRG	0.991	0.988	0.990	0.988	0.990
実行時間	TARGET	7.516	7.586	7.574	7.546	7.514
	SPLIT	48.940	63.51	74.682	81.63	87.3

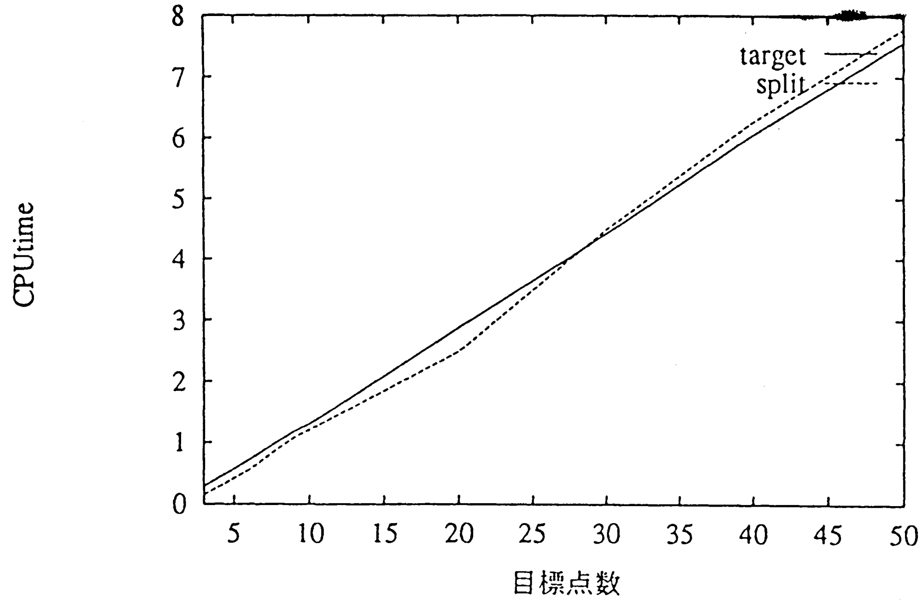

図 3:節点数= 50,　分割数= 3,　密度= 5%

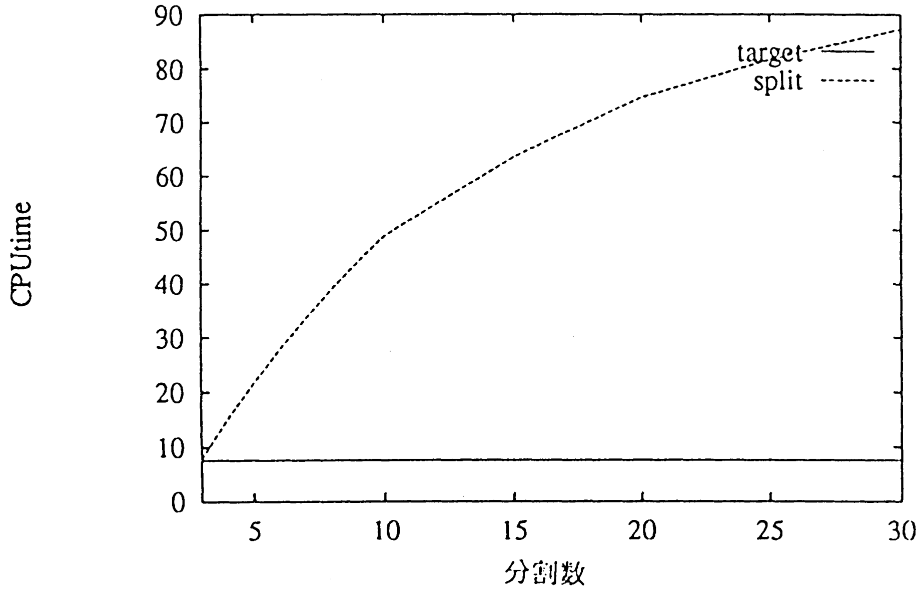

図 4:節点数= 50,　目標点数= 3,　密度= 5%

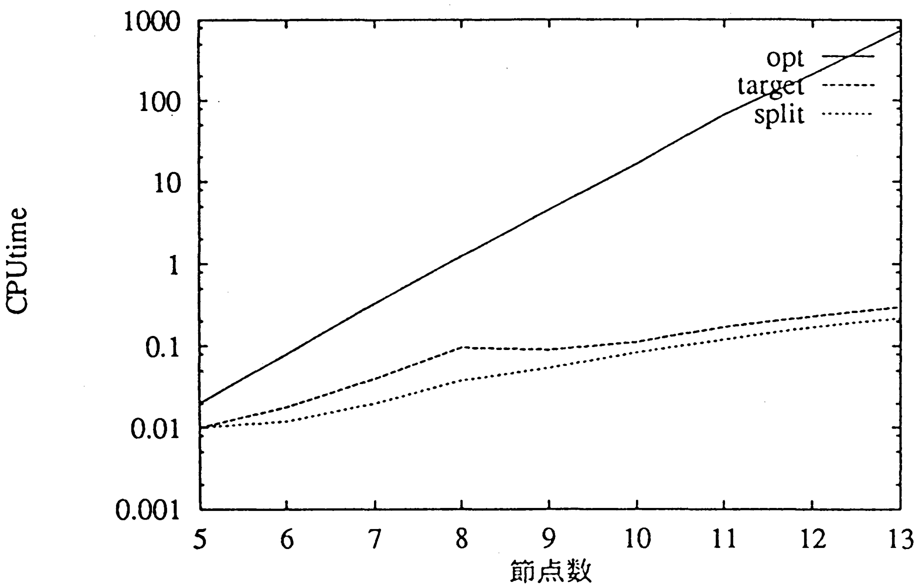

図 1:目標点数=節点数, 分割数= 3, 密度= 10%

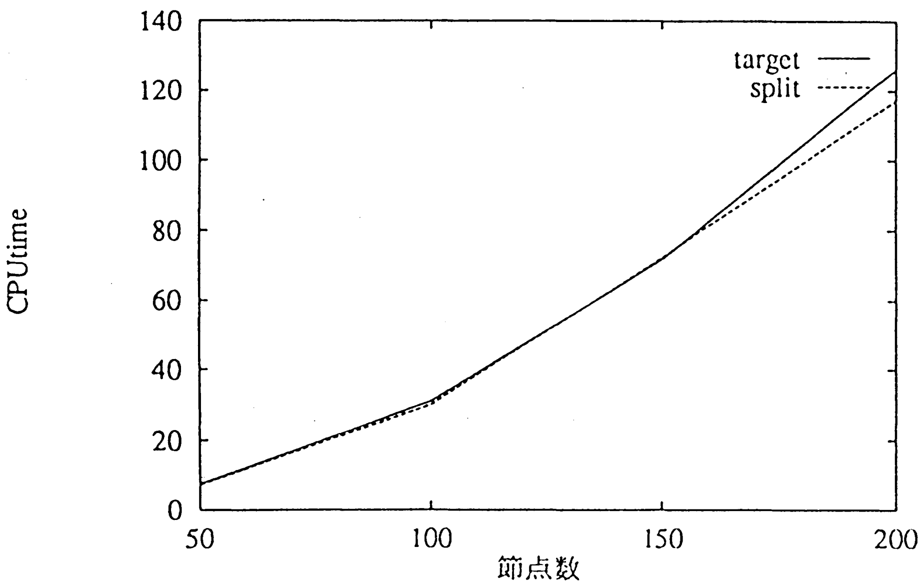

図 2:目標点数= 50, 分割数= 3, 密度= 5%